Amendments to the Specification:

Please amend the title as follows:

A1

HIGH-PERFORMANCE DIGITAL SIGNAL PROCESSOR (DSP) ~~RISC DSP~~ HAVING RISC INSTRUCTION EXECUTION PIPELINES

Please replace the paragraph beginning at page 2, line 2, with the following amended paragraph:

A2

Systems and method consistent with the present invention provide for an alternate DSP architecture configuration that allows for more efficient implementation of DSP **[[Dsp]]** algorithms and use of CPU resources.

Please replace the paragraph beginning at page 5, line 10, with the following amended paragraph:

A3

Fig. 1 shows an example of a DSP 100 having an architecture consistent with this invention. DSP 100 includes a Coprocessor 0 (CP0) 110, a Register File Arithmetic Logic Unit (RALU) 120, instructions memory and issue logic 130, data memory 140, and LBC 150. IADDR (instruction address b) bus 160 links Coprocessor **[[coprocessor]]** 0 110, instruction memory and issue logic 130, and LBC 150. DADDR (data address) bus 170 links RALU 120, data memory 140, and LBC 150. Instruction pathways INSTA 163 and INSTB 166 connect between CP0 110, RALU 120, instruction memory and issue logic 130, and LBC 150. DBUS (data bus) 175 connects between CP0 110, RALU 120, data memory 140, and LBC 150.

Please replace the paragraph beginning at page 5, line 18, with the following amended paragraph:

A4

CI (customer interface)180 acts as an interface to a customer coprocessor and connects to INSTA pathway 163 and DBUS 175. LBC 150 also interfaces to DI (data in) bus 192, DO (data out) bus 194, address bus 198, **[[196,]]** and control bus 196. **[[198.]]**

Please replace the paragraph beginning at page 8, line 13, with the following amended paragraph:

A5

Decoding register 220 belongs to instruction analysis logic 240, which, along with the Instruction Select logic, is implemented in the D-Stage. Instruction analysis logic 240 generates five key signals: 0eA, leA, 0eB, 1eB and 1d0. 0eA indicates whether I0 can execute in Pipe A. This decision is based on the I0 opcode and the resources available in Pipe A. For example, if I0 is an ADD and Pipe A can execute an ADD, then 0eA = 1; if I0 is a MAC and Pipe A has no MAC then 0eA = 0. The complexity of this logic can be minimized by careful encoding operations and by careful partitioning of operations between the two Pipes. One simplification could be to partition the DSP partition so that the Pipe B instruction, IB_S_R, is routed only to RALU 120 (Fig. 1).

Please replace the paragraph beginning at page 9, line 1, with the following amended paragraph:

A6

A special signal 1d0 indicates that both I0, I1 are valid and I1 (the higher logical order) **[[and]]** depends on I0. This will occur if the result of I0 is used as a source by I1. In this case, only I0 will issue. Analysis of the other signals is shown below:

Signal	Meaning
0eA	I0 is valid, I0 can be executed in Pipe A
1eA	I1 is valid, I1 can be executed in Pipe A
0eB	I0 is valid, I0 can be executed in Pipe B
1eB	I1 is valid, I1 can be executed in Pipe B
1d0	I0, I1 are valid and depends on I0: - source of I1 = destination **[[dest]]** of I0, or - I1 updates register file and I0 post-modifies load/store pointer

Please replace the paragraph beginning at page 10, line 1, with the following amended paragraph:

A7

Instruction select logic 250 controls output multiplexers 268 and 270 **[[272, 274]]** according to Table 300 in Fig. 3. Table 300 also specifies the update to the Valid register as a consequence of the issue decision.

Please replace the paragraph beginning at page 12, line 14, with the following amended paragraph:

A8

All ALU operations are available in both Pipe A and Pipe B. DSP extensions to memory addressing, such as pointer post-modification and circular buffer addressing described below, are preferably unique to Pipe A. Also, coprocessor operations and all "sequencing control instructions" (branches, jumps) are unique to Pipe A. As a result, Pipe B instructions are not routed to the coprocessors. This is shown in Fig. 4 with ALU B being connected to Custom Engine Interface (CEI) 450 and dual MAC 440, **[[44,]]** which preferably resides in RALU 120 (Fig. 1). CEI 450 is optionally available for customer proprietary operations only in Pipe B. This feature allows the customer extensions to maintain high throughput since they can dual-issue with Load and Store instructions which issue to Pipe A.

Please replace the paragraph beginning at page 21, line 11, with the following amended paragraph:

A9

The DSP supports three circular buffers. To initialize the circular buffers, MTALU (Move to ALU) instructions are used to set the twinword start addresses cbs 0 1010, cbs 1 1013, and cbs 2 1016 [31:3] and the twinword end addresses cbe 0 1020, cbe 1 1023, and cbe 2 **[[3]]** 1026 [31:3]. Circular buffers are only used when memory pointers are post-modified, and consist of an integral number of twinwords.